Game Theory Explorer - Software for the Applied Game Theorist

Bernhard von Stengel

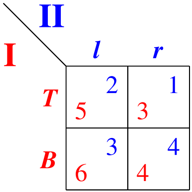
Department of Mathematics
London School of Economics

July 2013

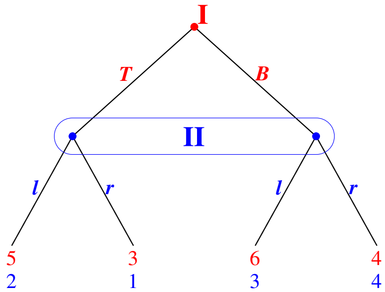
Overview

Explain and demonstrate GTE (Game Theory Explorer), open-source software, **under development**, for creating and analyzing non-cooperative **games**

in strategic form:



and extensive form:



Intended users

Applied game theorists:

- experimental economists (analyze game before running experiment)
- game-theoretic modelers in biology, political science, . . .
- in general: non-experts in equilibrium analysis

⇒ design goal: **ease of use**

Researchers in game theory:

- testing conjectures about equilibria
- as contributors: designers of game theory algorithms

History: Gambit

GTE now part of the **Gambit** open-source software development,
`http://www.gambit-project.org`

2011 and 2012 supported by Google Summer of Code (GSoC)

Gambit software started ∼1990 with **Richard McKelvey** (Caltech) to analyze games for **experiments**, developed since 1994 with **Andy McLennan** into C++ package, since 2001 maintained by **Ted Turocy** (Norwich, UK).

- Gambit must be **installed** on PC/Mac/Linux, with GUI (graphical user interface) using platform-independent wxWidget
- has collection of algorithms for computing Nash equilibria
- offers **scripting language**, now developed using Python

Features of GTE

GTE independent **browser-based** development:

- no software installation needed, low barrier to entry
- nicer GUI than Gambit
- export to graphical formats

Features of GTE

GTE independent **browser-based** development:

- no software installation needed, low barrier to entry
- nicer GUI than Gambit
- export to graphical formats

Disadvantages:

- manual storing / loading of files for security reasons
- long computations require local server installation (same GUI)

Features of GTE

GTE independent **browser-based** development:

- no software installation needed, low barrier to entry
- nicer GUI than Gambit
- export to graphical formats

Disadvantages:

- manual storing / loading of files for security reasons
- long computations require local server installation (same GUI)

Contributors:

David Avis (**lrs**), Rahul Savani (PhD 2006), Mark Egesdal (2011), Alfonso Gomez-Jordana, Martin Prause (**GSoC 2011, 2012**)

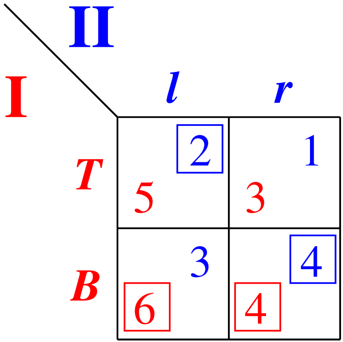
Example of a game

2 $\times$ **2** game in strategic form:

I \ II	l	r
T	5, 2	3, 1
B	6, 3	4, 4

Example of a game

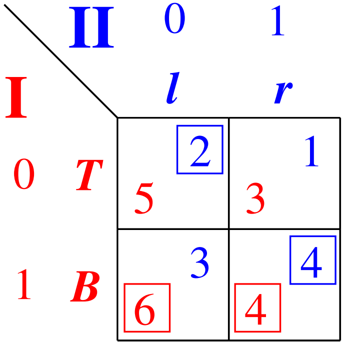
2 × **2** game in strategic form:



with pure best responses

Example of a game

2 × **2** game in strategic form:



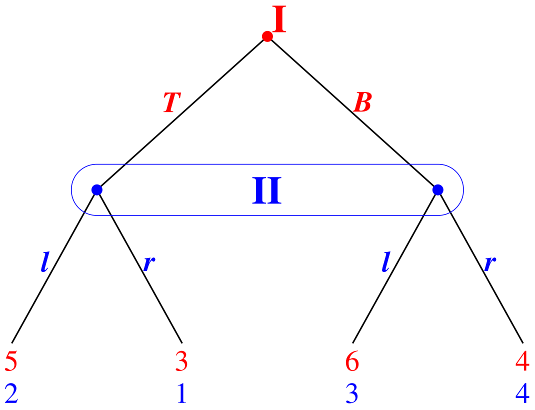
with pure best responses

and equilibrium probabilities

Extensive (= tree) form of the game

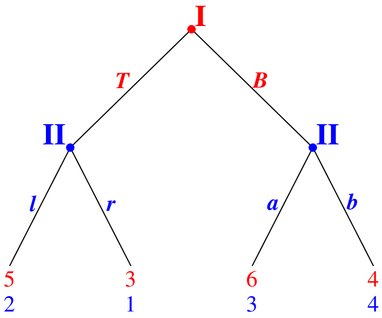
Players move sequentially,

information sets show **lack of information** about game state:



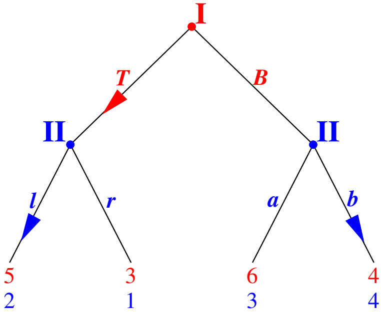
Commitment (leadership) game

Changed game when player **I** can commit:



Commitment (leadership) game

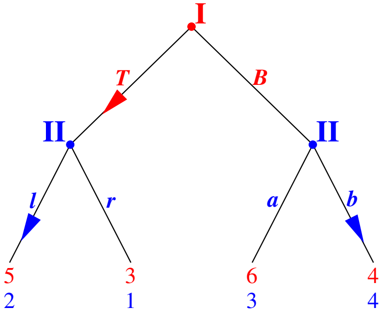
Changed game when player **I** can commit:



Subgame perfect equilibrium: **(*T*, *l*-*b*)**

Commitment (leadership) game

Changed game when player **I** can commit:

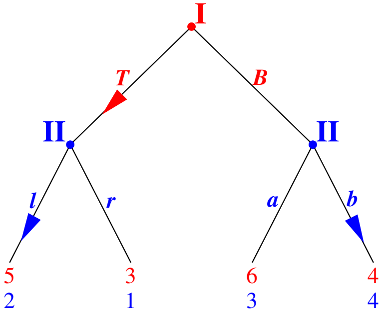


I \ II	*l–a*	*l–b*	*r–a*	*r–b*
T	5, 2	5, 2	3, 1	3, 1
B	6, 3	4, 4	6, 3	4, 4

Subgame perfect equilibrium: **(*T*, *l*-*b*)**

Commitment (leadership) game

Changed game when player **I** can commit:

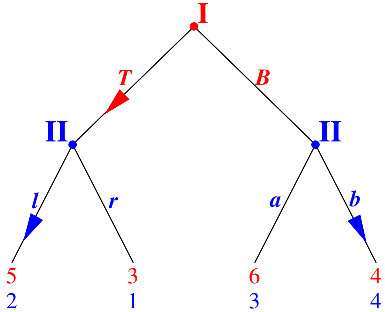


I \ II	l–a	l–b	r–a	r–b
T	5, 2	5, 2	3, 1	3, 1
B	6, 3	4, 4	6, 3	4, 4

Subgame perfect equilibrium: **(*T*, *l*-*b*)**

Commitment (leadership) game

Changed game when player **I** can commit:

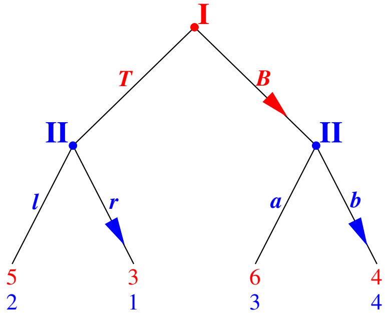


		II: 0 *l–a*	1 *l–b*	0 *r–a*	0 *r–b*
I: 1	*T*	5, 2	5, 2	3, 1	3, 1
0	*B*	6, 3	4, 4	6, 3	4, 4

Subgame perfect equilibrium: **(*T*, *l*-*b*)**

Commitment (leadership) game

Changed game when player **I** can commit:



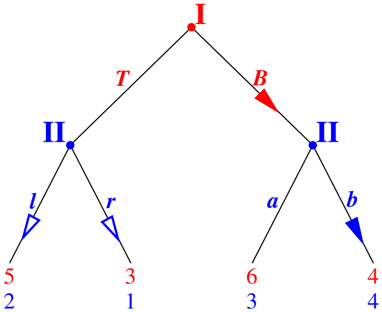
I \ II		0	0	0	1
		l–a	*l–b*	*r–a*	*r–b*
0	*T*	5, 2	5, 2	3, 1	3, 1
1	*B*	6, 3	4, 4	6, 3	4, 4

Subgame perfect equilibrium: **(*T*, *l*-*b*)**

Other equilibria: **(*B*, *r*-*b*)**

Commitment (leadership) game

Changed game when player **I** can commit:



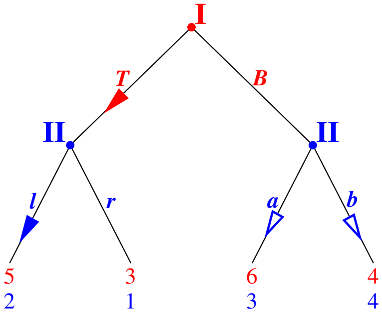
I \ II		0	1/2	0	1/2
		l–a	*l–b*	*r–a*	*r–b*
0	*T*	5, 2	5, 2	3, 1	3, 1
1	*B*	6, 3	4, 4	6, 3	4, 4

Subgame perfect equilibrium: **(*T*, *l*-*b*)**

Other equilibria: **(*B*, *r*-*b*), (*B*, $\frac{1}{2}$*l*-*b* $\frac{1}{2}$*r*-*b*)**

Commitment (leadership) game

Changed game when player **I** can commit:



I \ II		1/2	1/2	0	0
		l–a	*l–b*	*r–a*	*r–b*
1	*T*	5, 2	5, 2	3, 1	3, 1
0	*B*	6, 3	4, 4	6, 3	4, 4

Subgame perfect equilibrium: **(*T*, *l*-*b*)**

Other equilibria: **(*B*, *r*-*b*)**, **(*B*, $\frac{1}{2}$*l*-*b* $\frac{1}{2}$*r*-*b*)**, **(*T*, $\frac{1}{2}$*l*-*a* $\frac{1}{2}$*l*-*b*)**

GTE output for the commitment game

```
2 x 4 Payoff player 1          2 x 4 Payoff player 2
  l-a l-b r-a r-b                l-a l-b r-a r-b
T   5   5   3   3              T   2   2   1   1
B   6   4   6   4              B   3   4   3   4

EE = Extreme Equilibrium, EP = Expected Payoffs

Rational:
EE 1 P1: (1) 0 1 EP= 4 P2: (1)   0 1/2 0 1/2 EP= 4
EE 2 P1: (1) 0 1 EP= 4 P2: (2)   0   0 0   1 EP= 4
EE 3 P1: (2) 1 0 EP= 5 P2: (3)   0   1 0   0 EP= 2
EE 4 P1: (2) 1 0 EP= 5 P2: (4) 1/2 1/2 0   0 EP= 2

Connected component 1:
{1}  x  {1, 2}

Connected component 2:
{2}  x  {3, 4}
```

Demonstration of GTE

Preceding games:

- **2** $\times$ **2** game in strategic form
- extensive form of that game
- commitment game, extensive and strategic form

Demonstration of GTE

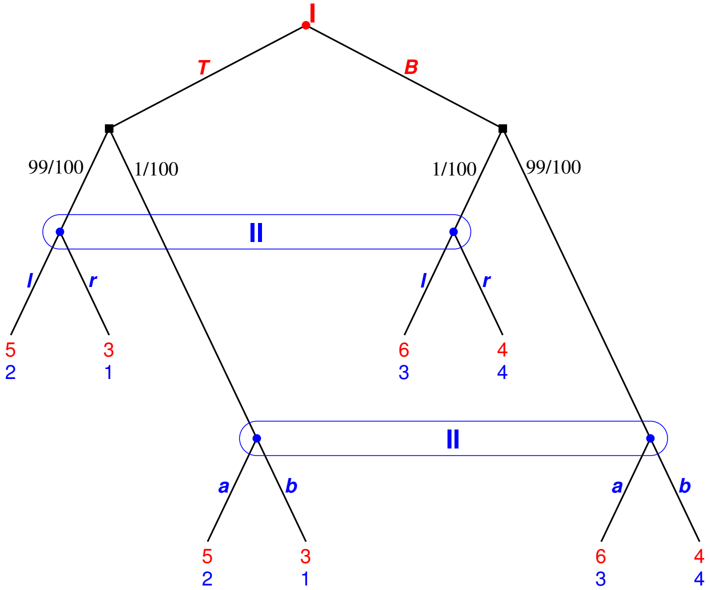
Preceding games:

- **2** $\times$ **2** game in strategic form
- extensive form of that game
- commitment game, extensive and strategic form

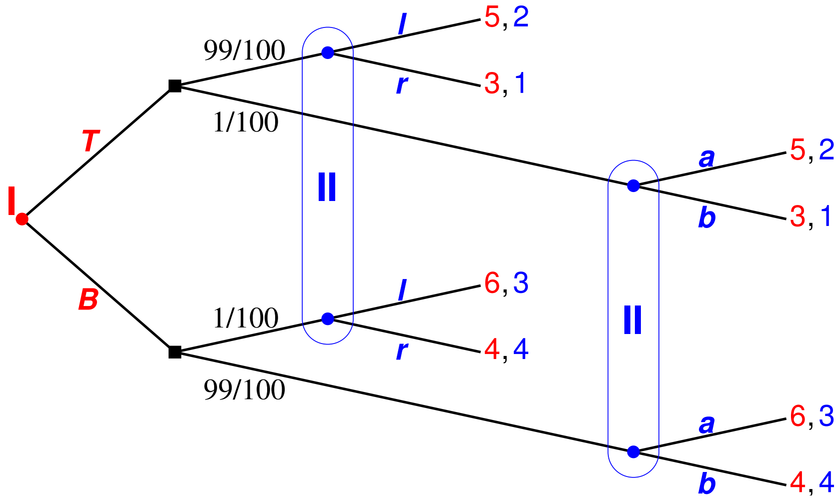
Next: create from scratch a more complicated extensive game

- imperfectly observable commitment

Game with imperfectly observable commitment



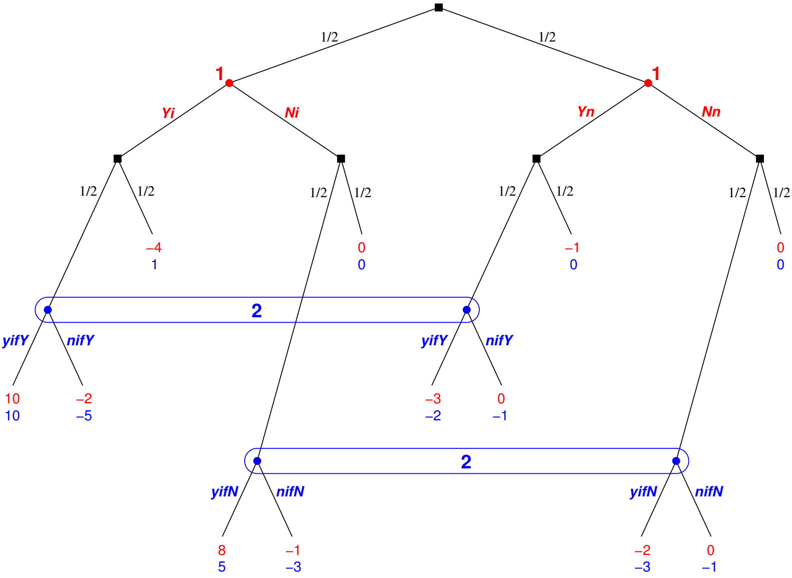
Game tree drawn left to right



GTE output for imperfectly observable commitment

```
2 x 4 Payoff player 1              2 x 4 Payoff player 2

  l-a     l-b     r-a r-b            l-a      l-b      r-a r-b
T   5 249/50 151/50   3            T   2 199/100 101/100   1
B   6 201/50 299/50   4            B   3 399/100 301/100   4

EE = Extreme Equilibrium, EP = Expected Payoffs

Decimal:
EE 1 P1: (1) 0.01 0.99 EP= 4.0102 P2: (1)      0 0.5102 0 0.4898 EP= 3.97
EE 2 P1: (2)    0  1.0 EP=    4.0 P2: (2)      0      0 0    1.0 EP=  4.0
EE 3 P1: (3) 0.99 0.01 EP= 4.9898 P2: (3) 0.4898 0.5102 0      0 EP= 2.01

Connected component 1:
{1}  x  {1}

Connected component 2:
{2}  x  {2}

Connected component 3:
{3}  x  {3}
```

More complicated signaling game, 5 equilibria



Some more strategic-form games

For studying more complicated games:

generate game matrices as text files, copy and paste into strategic-form input.

Future extension:

Automatic generation via command lines or “worksheets” for scripting, connection with Python and Gambit

GTE software architecture

Client (your computer with a browser):

- GUI: JavaScript (Flash's variant called ActionScript)
- store and load game described in XML format
- export to graphic formats (.png or XFIG $\rightarrow$ EPS, PDF)
- for algorithm: send XML game description to server

GTE software architecture

Client (your computer with a browser):

- GUI: JavaScript (Flash's variant called ActionScript)
- store and load game described in XML format
- export to graphic formats (.png or XFIG $\rightarrow$ EPS, PDF)
- for algorithm: send XML game description to server

Server (hosting client program and equilibrium solvers):

- converts XML to Java data structure (similar to GUI)
- solution algorithms as binaries (e.g. C program **lrs**), send results as text back to client

$\Rightarrow$ cannot use restrictive public servers like "Google App Engine"

High usage of computation resources

Finding all equilibria takes exponential time

$\Rightarrow$ for large games, server should run on your computer, not a public one

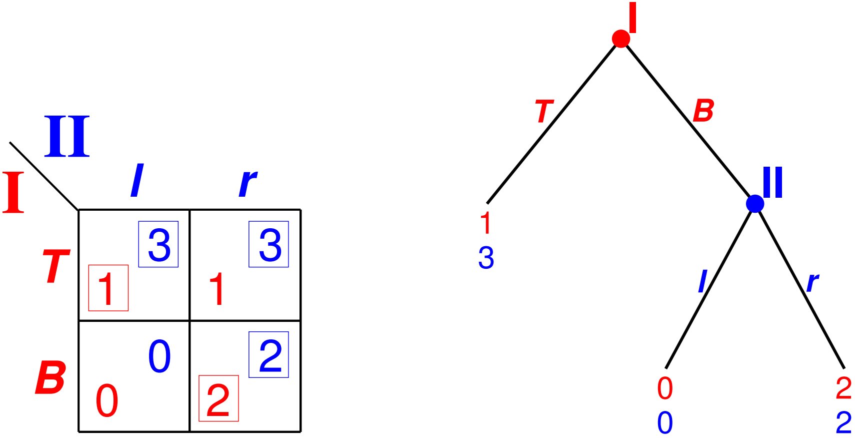
achieved by local server installation (“Jetty”), requires installation, but offers same user interface.

Algorithm: Finding all equilibria

For two-player games in strategic form, all Nash equilibria can be found as follows:

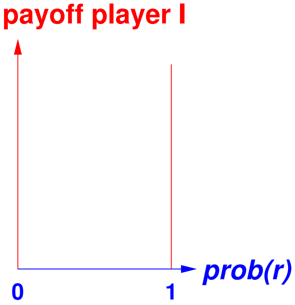
- payoffs define inequalities for “best response polyhedra”
- compute **all vertices** of these polyhedra (using **lrs** by David Avis, requires arbitrary precision integers)
- match vertices for **complementarity** (LCP)
- find maximal **cliques** of matching vertices for equilibrium **components**

Example



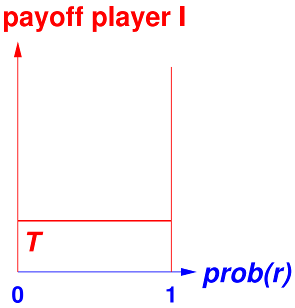
Best response polyhedron of player I

I \ II	*l*	*r*
T	1, 3	1, 3
B	0, 0	2, 2

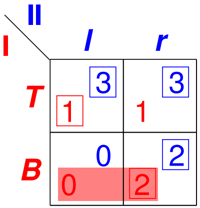


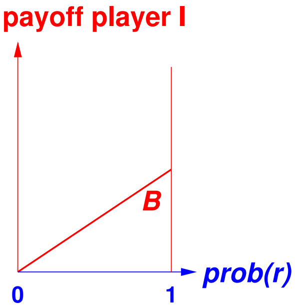
Best response polyhedron of player I

I \ II	*l*	*r*
T	1, 3	1, 3
B	0, 0	2, 2



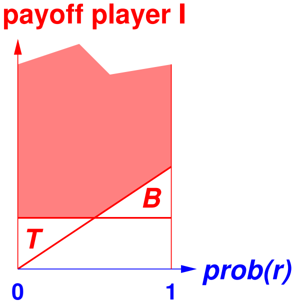
Best response polyhedron of player I





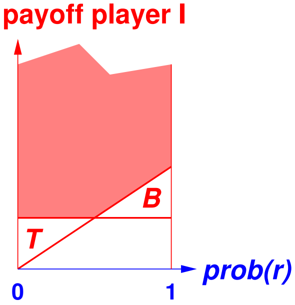
Best response polyhedron of player I

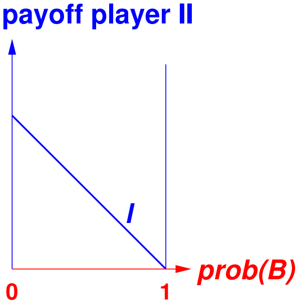
I \ II	*l*	*r*
T	1, 3	1, 3
B	0, 0	2, 2



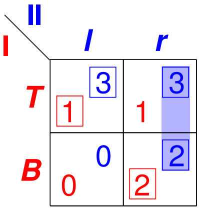
Best response polyhedron of player II

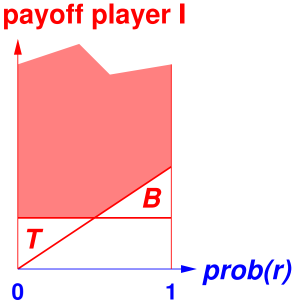
I \ II	*l*	*r*
T	1, 3	1, 3
B	0, 0	2, 2

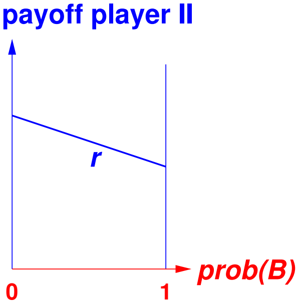




Best response polyhedron of player II

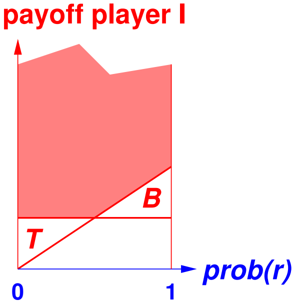


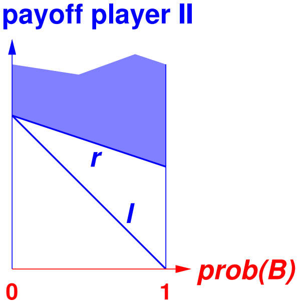




Best response polyhedron of player II

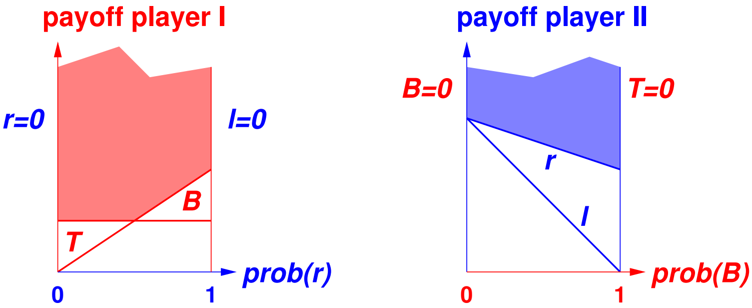
I \ II	l	r
T	1, 3	1, 3
B	0, 0	2, 2



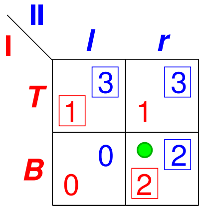


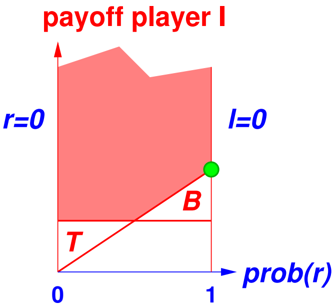
Label with best responses and unplayed strategies

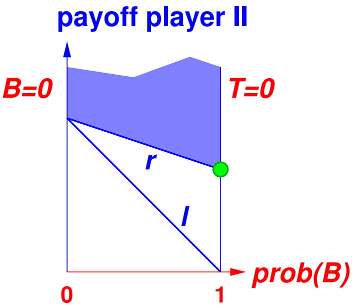
I \ II	*l*	*r*
T	1, 3	1, 3
B	0, 0	2, 2



Equilibrium = **all** labels ***T***, ***B***, ***l***, ***r*** present

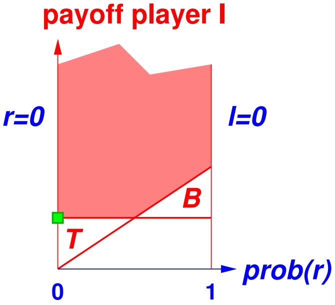


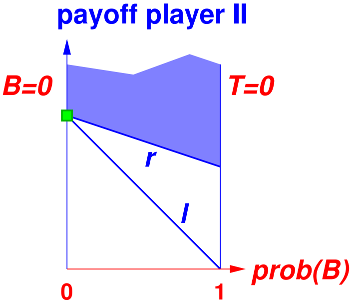




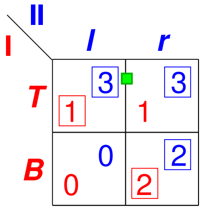
Equilibrium with multiple label *r* (degeneracy)

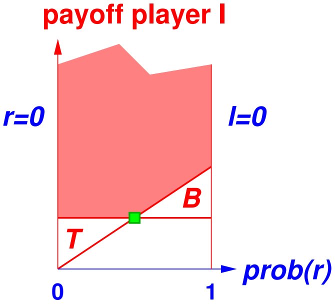
I \ II	*l*	*r*
T	1, 3	1, 3
B	0, 0	2, 2

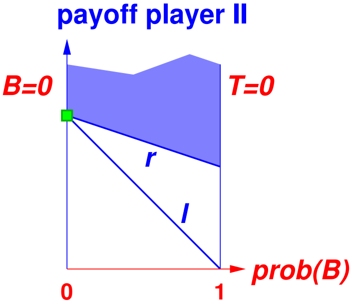




Equilibrium with multiple label *B* (degeneracy)

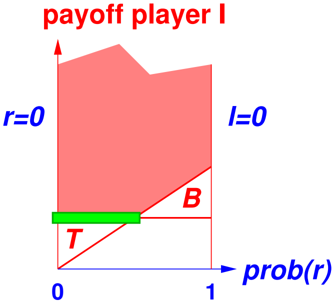


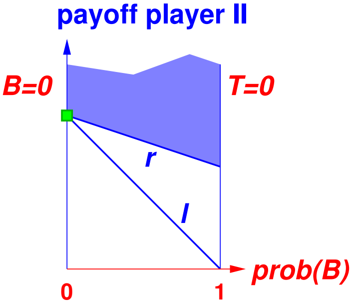




⇒ equilibrium component with labels *T* and *B*, *l*, *r*

I \ II	*l*	*r*
T	I: 1, II: 3	I: 1, II: 3
B	I: 0, II: 0	I: 2, II: 2





Equilibrium components via cliques

In degenerate games (= vertices with zero basic variables, occur for game trees), get convex combinations of “exchangeable” equilibria. Recognized as **cliques** of matching vertex pairs:

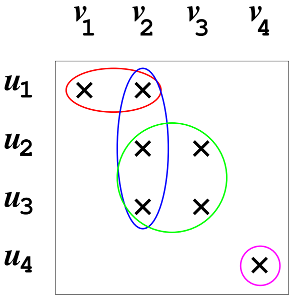
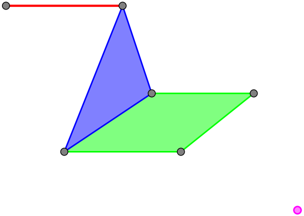


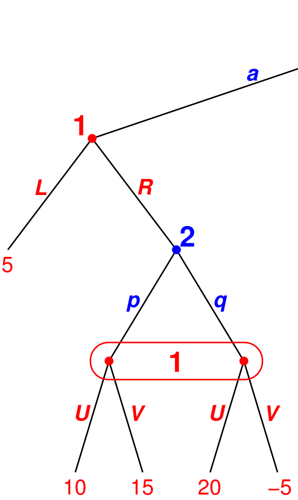
table of extreme equilibria



geometry

Algorithm: Sequence form for game trees

Example of game tree:



Exponentially large strategic form

Strategy of a player:
specifies a move for every information set of that player
(except for unspecified moves $*$ at unreachable information sets)

$\Rightarrow$ **exponential** number of strategies

	$ap*$	$aq*$	$b**$	$c*s$	$c*t$	$d**$
$L*C$	5	5	10	20	50	5
$L*D$	5	5	20	30	15	5
RUC	10	20	10	20	50	5
RUD	10	20	20	30	15	5
RVC	15	–5	10	20	50	5
RVD	15	–5	20	30	15	5

Sequences instead of strategies

Sequence specifies moves only along **path** in game tree

$\Rightarrow$ **linear** number of sequences, sparse payoff matrix $\boldsymbol{A}$

	$\emptyset$	a	b	c	d	ap	aq	cs	ct
$\emptyset$					5				
L		5							
R									
RU						10	20		
RV						15	–5		
C			10					20	50
D			20					30	15

Expected payoff $\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y}$, play rows with $\boldsymbol{x} \geq \boldsymbol{0}$ subject to $\boldsymbol{E}\boldsymbol{x} = \boldsymbol{e}$,

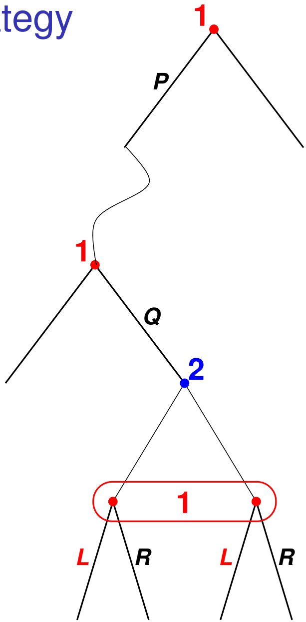
play columns with $\boldsymbol{y} \geq \boldsymbol{0}$ subject to $\boldsymbol{F}\boldsymbol{y} = \boldsymbol{f}$.

Play as behavior strategy

Given: $\boldsymbol{x} \geq \mathbf{0}$ with $\boldsymbol{E}\boldsymbol{x} = \boldsymbol{e}$.

Move $\boldsymbol{L}$ is last move of **unique** sequence, say $\boldsymbol{PQL}$, where one row of $\boldsymbol{E}\boldsymbol{x} = \boldsymbol{e}$ says $x_{PQL} + x_{PQR} = x_{PQ}$

$$\Rightarrow \quad \text{behavior-probability}(L) = \frac{x_{PQL}}{x_{PQ}}$$

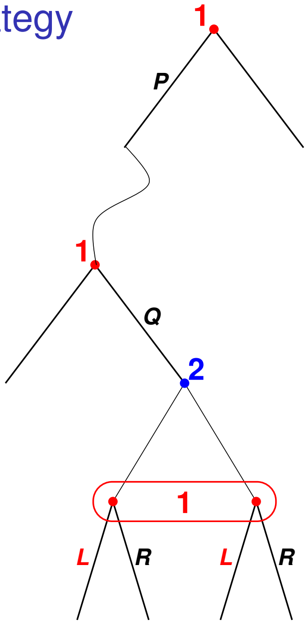


Play as behavior strategy

Given: $x \geq \mathbf{0}$ with $Ex = e$.

Move L is last move of **unique** sequence, say PQL, where one row of $Ex = e$ says $x_{PQL} + x_{PQR} = x_{PQ}$

$$\Rightarrow \quad \text{behavior-probability}(L) = \frac{x_{PQL}}{x_{PQ}}$$



Required assumption of **perfect recall** [Kuhn 1953, Selten 1975]:
Each node in an information set is preceded by same sequence, here PQ, of the player's **own** earlier moves.

Linear-sized sequence form

Input: Two-person game tree with perfect recall.

Theorem [Romanovskii 1962, vS 1996]

The equilibria of a **zero-sum** game are the solutions to a Linear Program (LP) of **linear** size in the size of the game tree.

Linear-sized sequence form

Input: Two-person game tree with perfect recall.

Theorem [Romanovskii 1962, vS 1996]

The equilibria of a **zero-sum** game are the solutions to a Linear Program (LP) of **linear** size in the size of the game tree.

Theorem [Koller/Megiddo/vS 1996, vS/Elzen/Talman 2002]

The equilibria of a **non-zero-sum** game are the solutions to a Linear Complementarity Problem (LCP) of linear size.

A sample equilibrium is found by **Lemke's algorithm**.

This algorithm mimicks the Harsanyi–Selten tracing procedure and finds a normal-form perfect equilibrium.

Planned Extensions

Improve and convert GUI to pure JavaScript (Flash is phased out)

Further solution algorithms:

- **EEE** [Audet/Hansen/Jaumard/Savard 2001], needs exact arithmetic
- Path-following algorithms (Lemke-Howson, variants of Lemke)
- ***n***-player games: simplicial subdivision, polynomial inequalities

Scripting features:

- connect with Gambit and Python
- database of reproducible computational experiments

Implementation challenges

Demonstrating that an algorithm works (for a publication)

- does not usually create robust and easy-to-use software

Implementation challenges

Demonstrating that an algorithm works (for a publication)

- does not usually create robust and easy-to-use software

Who should write such software?

- MSc thesis: not enough time
- PhD thesis / research grant: not scientific enough
- ideal: researcher creating “showcase” of their work

  Example: Rahul Savani’s `http://banach.lse.ac.uk/`
- student programmers with **Google Summer of Code**: insecure funding, but helps find **volunteer** open-source contributors.

Summary

GTE – Game theory explorer

- helps **create**, **draw**, and **analyze** game-theoretic models
- user-friendly, browser-based, low barriers to entry
- open-source, work in progress, needs contributors

`https://github.com/gambitproject/gte/wiki/_pages`

Summary

GTE – Game theory explorer

- helps **create**, **draw**, and **analyze** game-theoretic models
- user-friendly, browser-based, low barriers to entry
- open-source, work in progress, needs contributors

`https://github.com/gambitproject/gte/wiki/_pages`

Thank you!